

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2

дисциплина: Архитектура компьютеров и операционных систем

Студент: Румянцев А.О

Группа: НБИбд-02-23

МОСКВА

2023 г.

Содержание

1.Цель работы.....	4
2.Задание.....	5
3.Теоретическое введение	6
Таблица 1	7
4. Выполнение лабораторной работы	8
4.1 Настройка Github	8
4.2 Базовая настройка Git.....	8
4.3 Создание SSH-ключа.....	9
4.4 Создание рабочего пространства и репозитория курса на основе шаблона.....	10
4.5 Создание репозитория курса на основе шаблона.....	11
4.6 Настройка каталога курса	14
4.7 Выполнение заданий для самостоятельной работы	15
5.Выводы	19
6.Список литературы	20

Список иллюстраций

Рисунок 1.....	8
Рисунок 2.....	8
Рисунок 3.....	8
Рисунок 4.....	8
Рисунок 5.....	9
Рисунок 6.....	9
Рисунок 7.....	9
Рисунок 8.....	9
Рисунок 9.....	10
Рисунок 10.....	10
Рисунок 11.....	10
Рисунок 12.....	11
Рисунок 13.....	11
Рисунок 14.....	12
Рисунок 15.....	12
Рисунок 16.....	13
Рисунок 17.....	13
Рисунок 18.....	14
Рисунок 19.....	14
Рисунок 20.....	14
Рисунок 21.....	14
Рисунок 22.....	14
Рисунок 23.....	15
Рисунок 24.....	15
Рисунок 25.....	15
Рисунок 26.....	16
Рисунок 27.....	16
Рисунок 28.....	16
Рисунок 29.....	16
Рисунок 30.....	16
Рисунок 31.....	17
Рисунок 32.....	17
Рисунок 33.....	17
Рисунок 34.....	18
Рисунок 35.....	18
Рисунок 36.....	18

1.Цель работы

Целью данной работы является изучить идеологию и применение средств контроля версий, а также приобрести практические навыки по работе с системой git.

2.Задание

1. Настройка github
2. Базовая настройка git
3. Создание SSH ключа
4. Создание рабочего пространства и репозитория курса на основе шаблона
5. Создание репозитория курса на основе шаблона
6. Настройка каталога курса

3. Теоретическое введение

Системы контроля версий (Version Control System, VCS) применяются при работе нескольких человек над одним проектом. Обычно основное дерево проекта хранится в локальном или удалённом репозитории, к которому настроен доступ для участников проекта. При внесении изменений в содержание проекта система контроля версий позволяет их фиксировать, совмещать изменения, произведённые разными участниками проекта, производить откат к любой более ранней версии проекта, если это требуется.

В классических системах контроля версий используется централизованная модель, предполагающая наличие единого репозитория для хранения файлов. Выполнение большинства функций по управлению версиями осуществляется специальным сервером. Участник проекта (пользователь) перед началом работы посредством определённых команд получает нужную ему версию файлов. После внесения изменений, пользователь размещает новую версию в хранилище. При этом предыдущие версии не удаляются из центрального хранилища и к ним можно вернуться в любой момент. Сервер может сохранять не полную версию изменённых файлов, а производить так называемую дельта-компрессию — сохранять только изменения между последовательными версиями, что позволяет уменьшить объём хранимых данных.

Системы контроля версий поддерживают возможность отслеживания и разрешения конфликтов, которые могут возникнуть при работе нескольких человек над одним файлом. Можно объединить (слить) изменения, сделанные разными участниками (автоматически или вручную), вручную выбрать нужную версию, отменить изменения вовсе или заблокировать файлы для изменения. В зависимости от настроек блокировка не позволяет другим пользователям получить рабочую копию или препятствует изменению рабочей копии файла средствами файловой системы ОС, обеспечивая таким образом, привилегированный доступ только одному пользователю, работающему с файлом.

Системы контроля версий также могут обеспечивать дополнительные, более гибкие функциональные возможности. Например, они могут поддерживать работу с несколькими версиями одного файла, сохраняя общую историю изменений до точки ветвления версий и собственные истории изменений каждой ветви. Кроме того, обычно доступна информация о том, кто из участников, когда и какие изменения вносил. Обычно такого рода информация хранится в журнале изменений, доступ к которому можно ограничить.

В отличие от классических, в распределённых системах контроля версий центральный репозиторий не является обязательным.

Среди классических VCS наиболее известны CVS, Subversion, а среди распределённых — Git, Bazaar, Mercurial. Принципы их работы схожи, отличаются они в основном синтаксисом используемых в работе команд.

Система контроля версий Git представляет собой набор программ командной строки. Доступ к ним можно получить из терминала посредством ввода команды `git` с различными опциями. Благодаря тому, что Git является распределённой системой контроля версий, резервную копию локального хранилища можно сделать простым копированием или архивацией.

Наиболее часто используемые команды git представлены в таблице 1:

Таблица 1

Команда	Описание
git init	Создание основного дерева репозитория
git pull	получение обновлений (изменений) текущего дерева из центрального репозитория
git push	отправка всех произведённых изменений локального дерева в центральный репозиторий
git status	просмотр списка изменённых файлов в текущей директории
git diff	просмотр текущих изменения
git add	добавить все изменённые и/или созданные файлы и/или каталоги
git add имена_файлов	добавить конкретные изменённые и/или созданные файлы и/или каталоги
git rm имена_файлов	удалить файл и/или каталог из индекса репозитория (при этом файл и/или каталог остаётся в локальной директории)
git commit -am 'Описание коммита'	сохранить все добавленные изменения и все изменённые файлы
git checkout -b имя_ветки	создание новой ветки, базирующейся на текущей
git checkout имя_ветки	переключение на некоторую ветку (при переключении на ветку, которой ещё нет в локальном репозитории, она будет создана и связана с удалённой)
git push origin имя_ветки	отправка изменений конкретной ветки в центральный репозиторий
git merge --no-ff имя_ветки	слияние ветки с текущим деревом
git branch -d имя_ветки	удаление локальной уже слитой с основным деревом ветки
git branch -D имя_ветки	принудительное удаление локальной ветки
git push origin :имя_ветки	удаление ветки с центрального репозитория

4. Выполнение лабораторной работы

4.1 Настройка Github

Создаю учетную запись на сайте GitHub. Далее я заполняю основные данные учетной записи и регистрирую аккаунт. Аккаунт создан (рис 1)

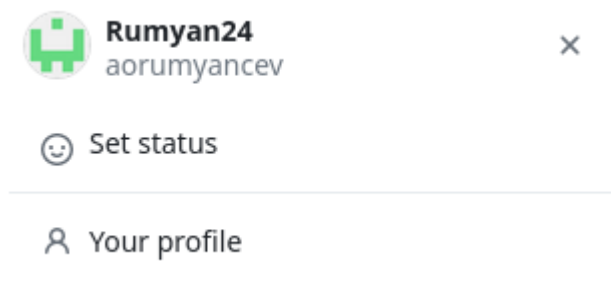


Рисунок 1

4.2 Базовая настройка Git

Запускаю виртуальную машину, затем в терминале задаю предварительную конфигурацию git. Ввожу команду `git config --global user.name ""`, указывая свое имя и команду `git config --global user.email "work@mail"`, указывая в ней электронную почту владельца, то есть мою (рис. 2).

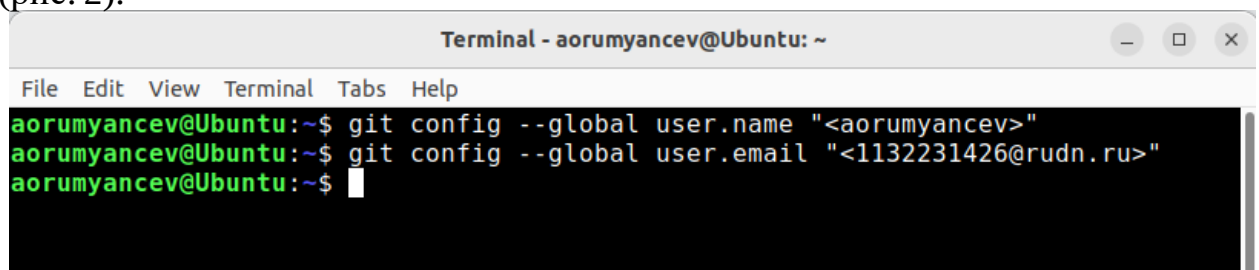


Рисунок 2

Настраиваю utf-8 в выводе сообщений git для корректного отображения символов(рис.3).

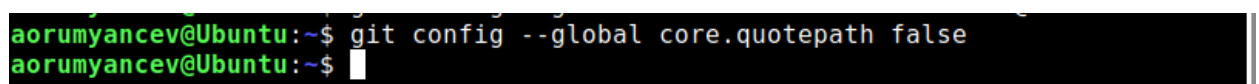


Рисунок 3

Задаю имя «master» для начальной ветки(рис.4).

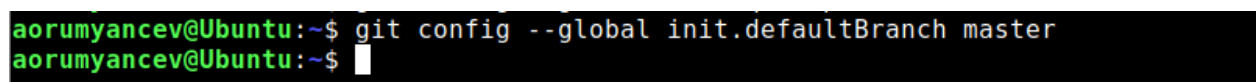


Рисунок 4

Задаю параметр autocrlf со значением input, (рис. 5).

```
aorumyancev@Ubuntu:~$ git config --global core.autocrlf input
aorumyancev@Ubuntu:~$
```

Рисунок 5

Задаю параметр `safecrlf` со значением `warn.`(рис. 6).

```
aorумыancev@Ubuntu:~$ git config --global core.safecrlf warn
aorумыancev@Ubuntu:~$
```

Рисунок 6

4.3 Создание SSH-ключа

Для последующей идентификации пользователя на сервере репозитория необходимо сгенерировать пару ключей (приватный и открытый). Для этого ввожу команду `ssh-keygen -C "Имя Фамилия, work@email"`, указывая имя владельца и электронную почту владельца (рис.7). Ключ автоматически сохранится в каталоге `~/.ssh/`.

```
Terminal - aorumyancev@Ubuntu: ~
File Edit View Terminal Tabs Help
aorumyancev@Ubuntu:~$ ssh-keygen -C "aorumyancev <1132231426@rudn.ru>"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/aorumyancev/.ssh/id_rsa): key
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in key
Your public key has been saved in key.pub
The key fingerprint is:
SHA256:bTGg/IKg5Z9sRhk5tLZu9PoiZaJMCcXm9GmX0w0o/GQ aorumyancev <1132231426@rudn.ru>
The key's randomart image is:
+---[RSA 3072]---+
| =..  . |
| =.ooE* + |
| . *=X + = |
| . = +.B + o |
| .... * S + |
| o .*oo o |
| o . +0 . |
| o .+.. |
| ..o. |
+----[SHA256]-----+
```

Рисунок 7

Копирую открытый ключ из директории, в которой он был сохранен, с помощью утилиты `xclip` (рис. 8).

```

aorumyancev@Ubuntu:~$ cat ~/key.pub | xclip -sel clip
aorumyancev@Ubuntu:~$

```

Рисунок 8

Открываю браузер, захожу на сайт GitHub. Открываю свой профиль и выбираю страницу «SSH and GPG keys». Нажимаю кнопку «New SSH key». Вставляю скопированный ключ в поле «Key». В поле Title указываю имя для ключа. Нажимаю «Add SSH-key», чтобы завершить добавление ключа (рис. 9,10,11)

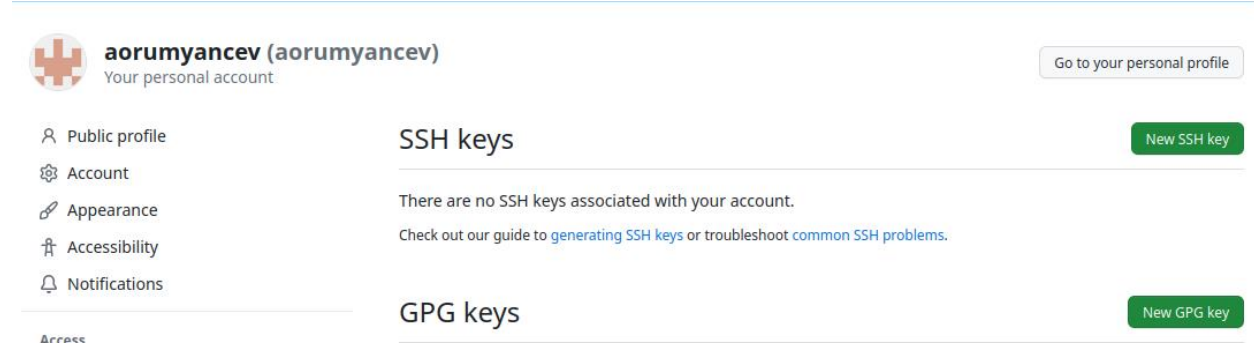


Рисунок 9

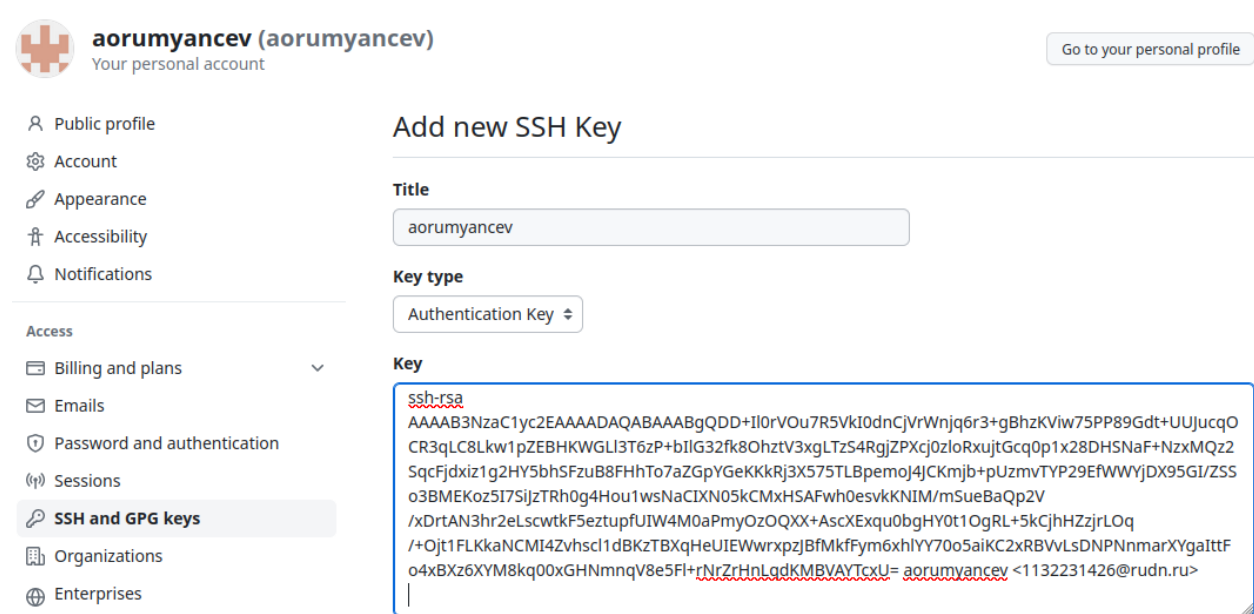


Рисунок 10

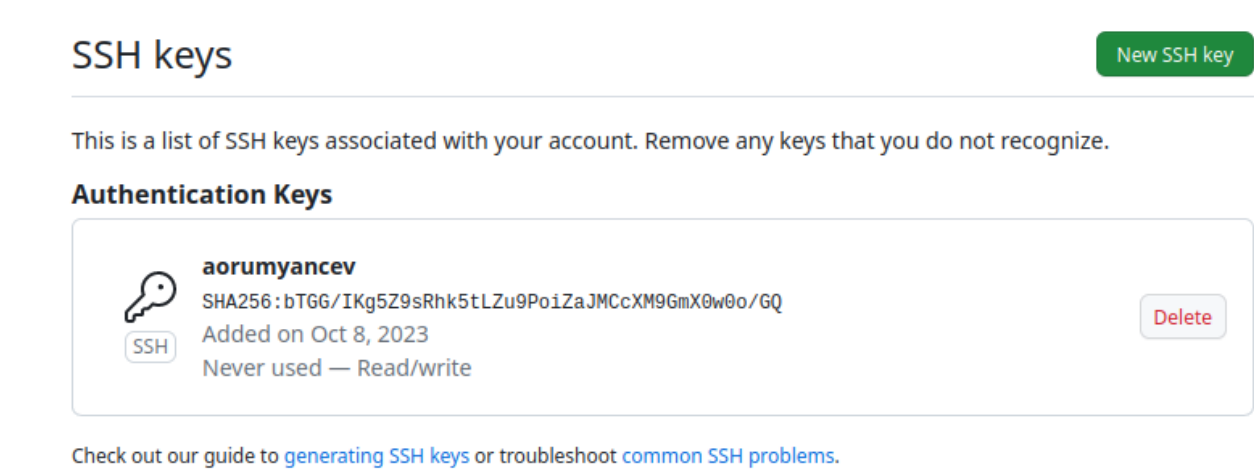


Рисунок 11

4.4 Создание рабочего пространства и репозитория курса на основе шаблона

Закрываю браузер, открываю терминал. Создаю директорию, рабочее пространство, с помощью утилиты `mkdir`, благодаря ключу `-p` создаю все директории после домашней `~/work/study/2023-2024/“Архитектура компьютера”` рекурсивно. Далее проверяю с помощью `ls`, действительно ли были созданы необходимые мне каталоги (рис. 12). (скриншот был сделан после клонирования т.к. забыл сделать во время выполнения работы)

```
aorumyancev@Ubuntu:~$ mkdir -p ~/work/study/2023-2024/"Архитектура компьютера"
aorumyancev@Ubuntu:~$ ls
arch-pc  Downloads  Music      snap       work
Desktop  key        Pictures   Templates  Документы
Documents key.pub    Public     Videos    Изображения
aorumyancev@Ubuntu:~$
```

Рисунок 12

4.5 Создание репозитория курса на основе шаблона

В браузере перехожу на страницу репозитория с шаблоном курса по адресу <https://github.com/yamadharm/course-directory-student-template>. Далее выбираю «Use this template», чтобы использовать этот шаблон для своего репозитория (рис.13).

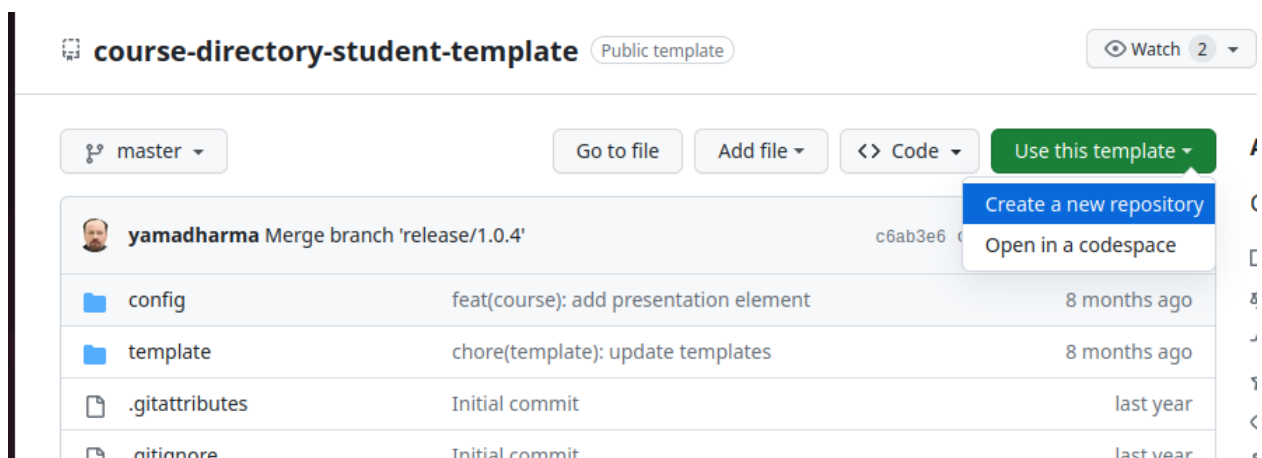


Рисунок 13

В открывшемся окне задаю имя репозитория (Repository name): study_2023–2024_arch-pc и создаю репозиторий, нажимаю на кнопку «Create repository»

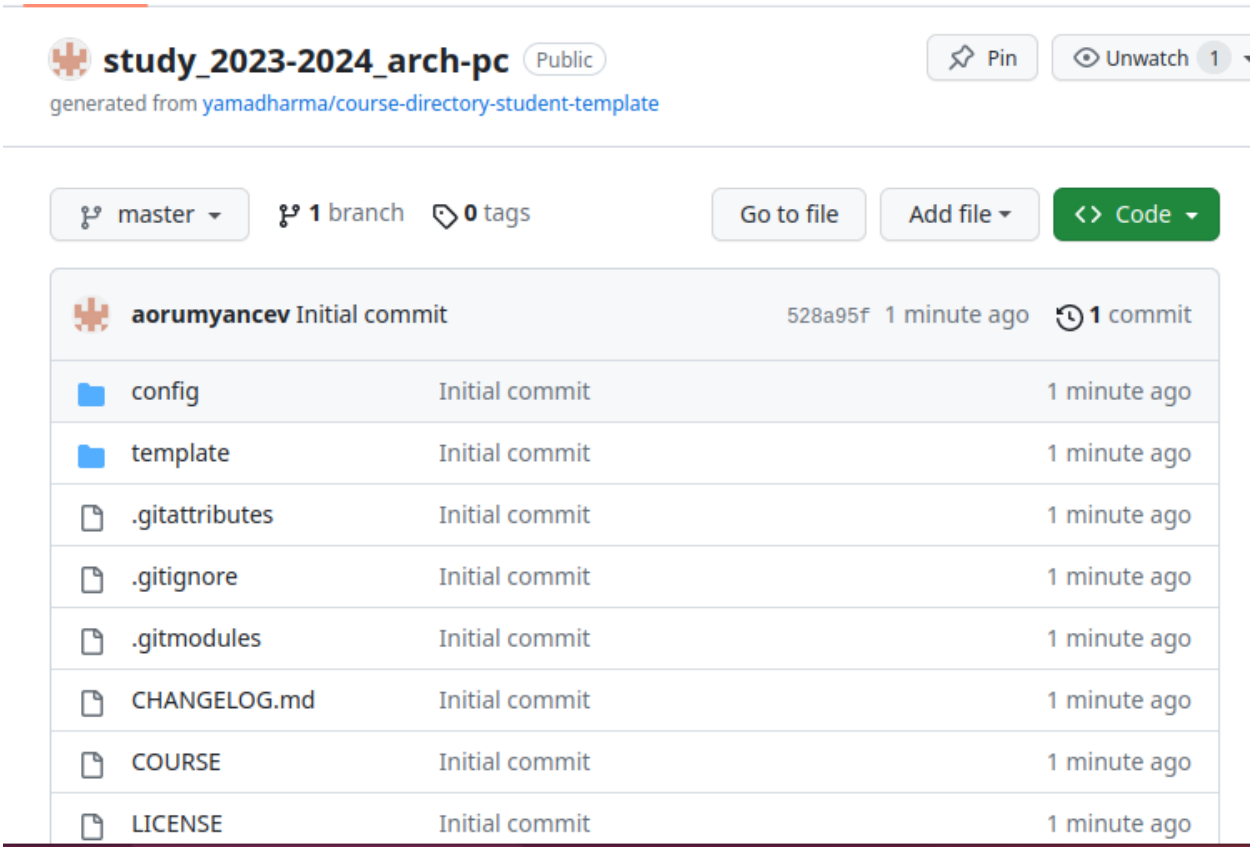


Рисунок 14

Через терминал перехожу в созданный каталог курса с помощью утилиты cd(рис. 15).

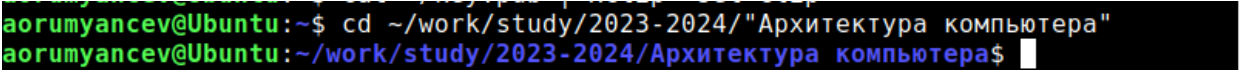


Рисунок 15

Копирую ссылку для клонирования на странице созданного репозитория, сначала перейдя в окно «code», далее выбрав в окне вкладку «SSH» (рис. 16).

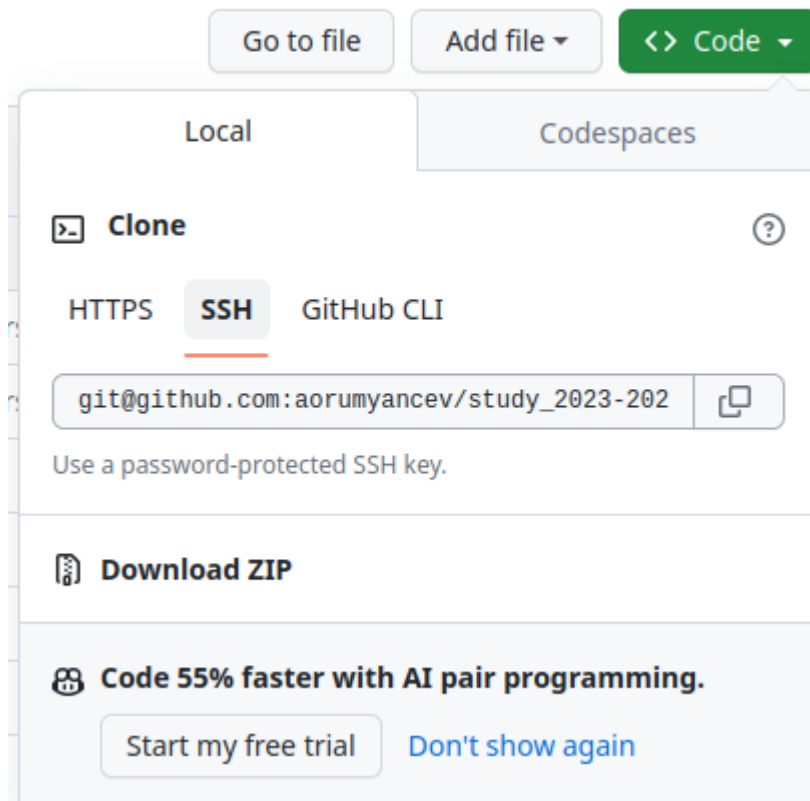


Рисунок 16

Клонирую созданный репозиторий с помощью команды

```
git clone --recursive git@github.com:/study_2023-2024_arch-pc.git
```

arch-pc (рис. 17).

```
aorumyancev@Ubuntu:~$ cd ~/work/study/2023-2024/Архитектура компьютера
aorumyancev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера$ git clone --recursive git@github.com:aorumyancev/study_2023-2024_arch-pc.git arch-pc
Cloning into 'arch-pc'...
remote: Enumerating objects: 27, done.
remote: Counting objects: 100% (27/27), done.
remote: Compressing objects: 100% (26/26), done.
remote: Total 27 (delta 1), reused 11 (delta 0), pack-reused 0
Receiving objects: 100% (27/27), 16.93 KiB | 16.93 MiB/s, done.
Resolving deltas: 100% (1/1), done.
Submodule 'template/presentation' (https://github.com/yamadharma/academic-presentation-markdown-template.git) registered for path 'template/presentation'
Submodule 'template/report' (https://github.com/yamadharma/academic-laboratory-report-template.git) registered for path 'template/report'
Cloning into '/home/aorumyancev/work/study/2023-2024/Архитектура компьютера/arch-pc/template/presentation'...
remote: Enumerating objects: 82, done.
remote: Counting objects: 100% (82/82), done.
remote: Compressing objects: 100% (57/57), done.
remote: Total 82 (delta 28), reused 77 (delta 23), pack-reused 0
Receiving objects: 100% (82/82), 92.90 KiB | 1001.00 KiB/s, done.
Resolving deltas: 100% (28/28), done.
Cloning into '/home/aorumyancev/work/study/2023-2024/Архитектура компьютера/arch-pc/template/report'...
remote: Enumerating objects: 101, done.
remote: Counting objects: 100% (101/101), done.
remote: Compressing objects: 100% (70/70), done.
remote: Total 101 (delta 40), reused 88 (delta 27), pack-reused 0
Receiving objects: 100% (101/101), 327.25 KiB | 2.14 MiB/s, done.
Resolving deltas: 100% (40/40), done.
Submodule path 'template/presentation': checked out 'b1be3800ee91f5809264cb755d316174540b753e'
Submodule path 'template/report': checked out '1d1b61dcac9c287a83917b82e3aef11a33b1e3b2'
aorumyancev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера$
```

Рисунок 17

4.6 Настройка каталога курса

Перехожу в каталог arch-pc с помощью утилиты cd (рис. 18).

```
aorumyancev@Ubuntu:~$ cd ~/work/study/2023-2024/"Архитектура компьютера"
aorumyancev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера$ cd ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc
aorumyancev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc$
```

Рисунок 18

Удаляю лишние файлы с помощью утилиты rm (рис. 19).

```
aorumyancev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ rm package.js
aorumyancev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc$
```

Рисунок 19

Создаю необходимые каталоги (рис. 20).

```
aorumyancev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ echo arch-pc
> COURSE
aorumyancev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ make
aorumyancev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc$
```

Рисунок 20

Отправляю созданные каталоги с локального репозитория на сервер: добавляю все созданные каталоги с помощью git add, комментирую и сохраняю изменения на сервере как добавление курса с помощью git commit (рис. 21).

```
aorumyancev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git add .
aorumyancev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git commit -am 'feat(main): make course structure'
[master 37bf1ef] feat(main): make course structure
199 files changed, 54725 insertions(+), 14 deletions(-)
create mode 100644 labs/README.md
create mode 100644 labs/README.ru.md
create mode 100644 labs/lab01/presentation/Makefile
create mode 100644 labs/lab01/presentation/image/kulyabov.jpg
create mode 100644 labs/lab01/presentation/presentation.md
create mode 100644 labs/lab01/report/Makefile
create mode 100644 labs/lab01/report/bib/cite.bib
create mode 100644 labs/lab01/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab01/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_eqnos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_fignos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_secnos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_tablenos.py
```

Рисунок 21

Отправляю все на сервер с помощью push(рис.22)

```
aorumyancev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc$ git push
Enumerating objects: 37, done.
Counting objects: 100% (37/37), done.
Delta compression using up to 6 threads
Compressing objects: 100% (29/29), done.
Writing objects: 100% (35/35), 342.13 KiB | 2.53 MiB/s, done.
Total 35 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:aorumyancev/study_2023-2024_arch-pc.git
528a95f..37bf1ef master -> master
aorumyancev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc$
```

Рисунок 22

Проверяю правильность выполнения работы на самом сайте
GitHub(рис. 23).

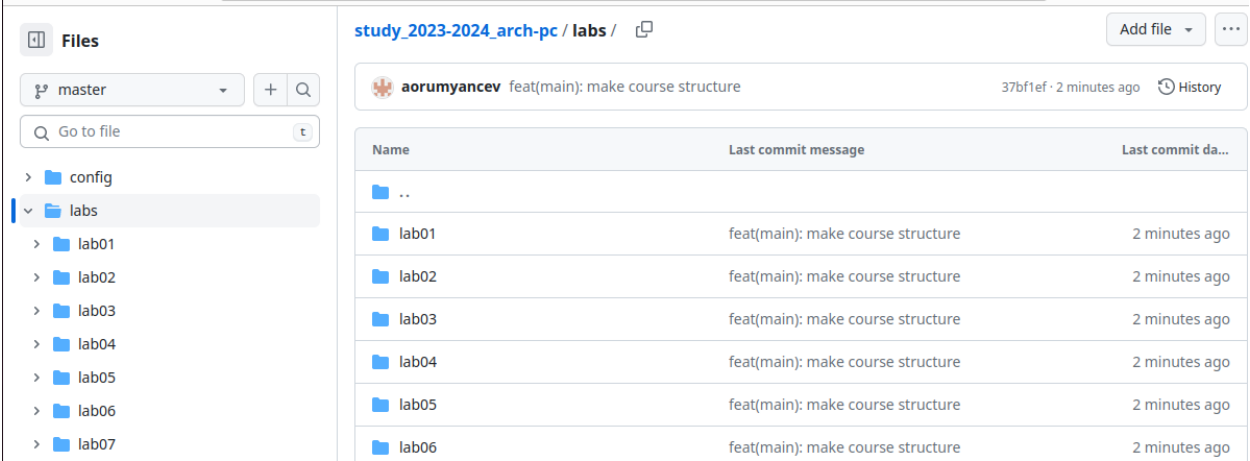


Рисунок 23

4.7 Выполнение заданий для самостоятельной работы

1. Перехожу в директорию labs/lab02/report с помощью утилиты cd. Создаю в каталоге файл для отчета по третьей лабораторной работе с помощью утилиты touch (рис. 24).

```
aorumyancev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab02/report$ touch Л02_Румянцев_отчёт
```

Рисунок 24

Оформить отчет я смогу в текстовом процессоре LibreOffice Writer, найдя его в меню приложений

Оформить отчет я смогу в текстовом процессоре LibreOffice Writer, найдя его в меню приложений (рис. 25).

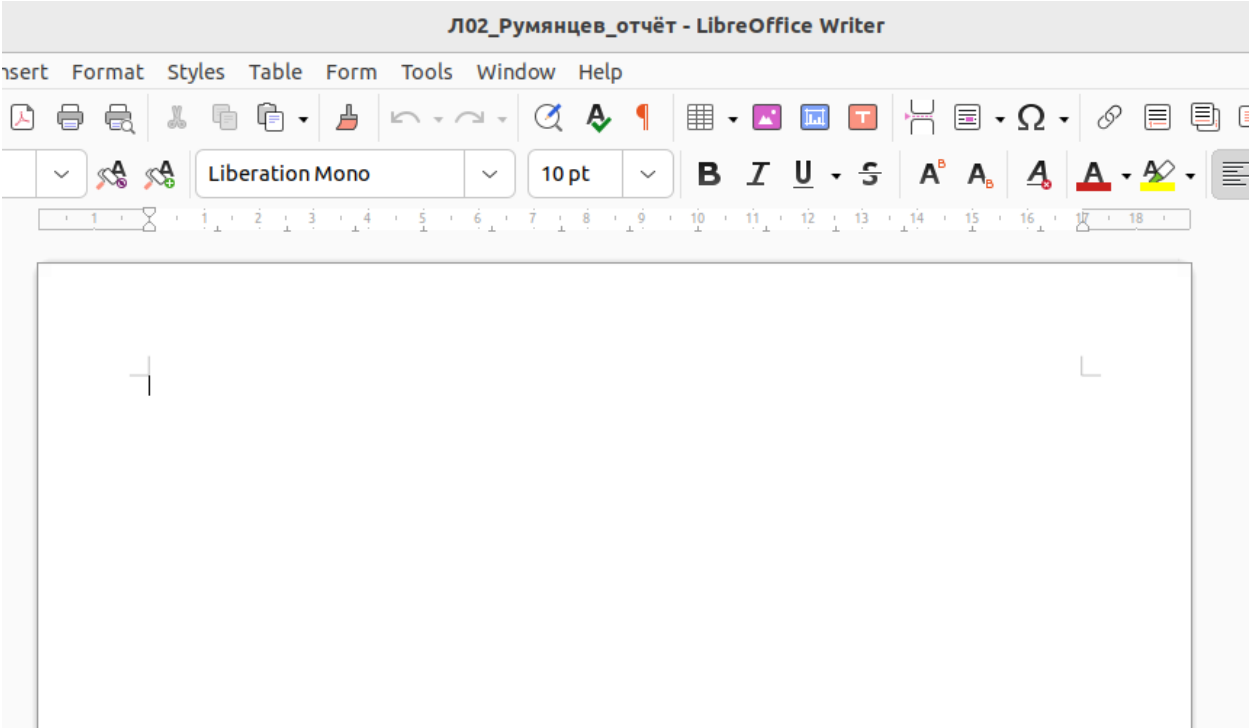


Рисунок 25

2. Перехожу из подкаталога lab02/report в подкаталог lab01/report с помощью утилиты cd (рис.26)

```
aorumyancev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab02/report$ cd ..
aorumyancev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab02$ cd ..
aorumyancev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs$ cd lab01
aorumyancev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab01$ cd report
aorumyancev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab01/report$
```

Рисунок 26

Проверяю местонахождение файлов с отчетами по первой лабораторной работе. Она должна быть в подкаталоге домашней директории «Downloads», для проверки использую команду ls (рис. 27)

```
aorumyancev@Ubuntu:~/Downloads$ ls
Л01_Румянцев_отчет..pdf
aorumyancev@Ubuntu:~/Downloads$
```

Рисунок 27

Копирую первую лабораторную с помощью утилиты cp и проверяю правильность выполнения команды cp с помощью ls (рис.28)

```
Terminal - aorumyancev@Ubuntu: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab01/report
File Edit View Terminal Tabs Help
aorumyancev@Ubuntu:~$ cp ~/Downloads/Л01_Румянцев_отчет..pdf /home/aorumyancev/work/study/2023-2024
aorumyancev@Ubuntu:~$ cd ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab01/report
aorumyancev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab01/report$ ls
bib image Makefile pandoc report.md Л01_Румянцев_отчет..pdf
aorumyancev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab01/report$
```

Рисунок 28

Добавляю файл Л01_Румянцев_отчет..pdf(рис.29).

```
aorumyancev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab01/report$ git add
Л01_Румянцев_отчет..pdf
aorumyancev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab01/report$
```

Рисунок 29

Сохраняю изменения на сервере командой git commit -m "...", поясняя, что добавил файлы.

То же самое делаю для отчета по третьей лабораторной работе: перехожу в директорию labs/lab02/report с помощью cd, добавляю с помощью git add нужный файл, сохраняю изменения с помощью git commit (рис. 30,31)

```
aorumyancev@Ubuntu:~$ cp ~/Downloads/Л01_Румянцев_отчет..pdf /home/aorumyancev/work/study/2023-2024
aorumyancev@Ubuntu:~$ cd ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab01/report
aorumyancev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab01/report$ ls
bib image Makefile pandoc report.md Л01_Румянцев_отчет..pdf
aorumyancev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab01/report$ git add
Л01_Румянцев_отчет..pdf
aorumyancev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab01/report$ git com
mit -m "Add existing files"
[master fc8f244] Add existing files
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 labs/lab01/report/Л01_Румянцев_отчет..pdf
```

Рисунок 30


```
aorumyancev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab02/report$ ls
bib image Makefile pandoc report.md Л02 Румянцев отчёт
aorumyancev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab02/report$ git add
Л02 Румянцев отчёт
aorumyancev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab02/report$ git com
mit -m "Add existing files"
[master 1126aac] Add existing files
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 labs/lab02/report/Л02 Румянцев отчёт
aorumyancev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab02/report$
```


Рисунок 31

Отправляю в центральный репозиторий сохраненные изменения командой `git push -f origin master` (рис.32)

```
aorumyancev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab02/report$ git pus
h -f origin master
Enumerating objects: 17, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 6 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (11/11), 1.56 MiB | 2.32 MiB/s, done.
Total 11 (delta 5), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (5/5), completed with 2 local objects.
To github.com:aorumyancev/study_2023-2024_arch-pc.git
 37bf1ef..1126aac master -> master
aorumyancev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab02/report$
```

Рисунок 32

Проверяю на сайте GitHub правильность выполнения заданий. Вижу, что пояснение к совершенным действиям отображается (рис.33,34)

 **aorumyancev** Add existing files 1126aac








Name	Last commit message
 ..	
 lab01	Add existing files
 lab02	Add existing files
 lab03	feat(main): make course structure
 lab04	feat(main): make course structure
 lab05	feat(main): make course structure
 lab06	feat(main): make course structure

Рисунок 33

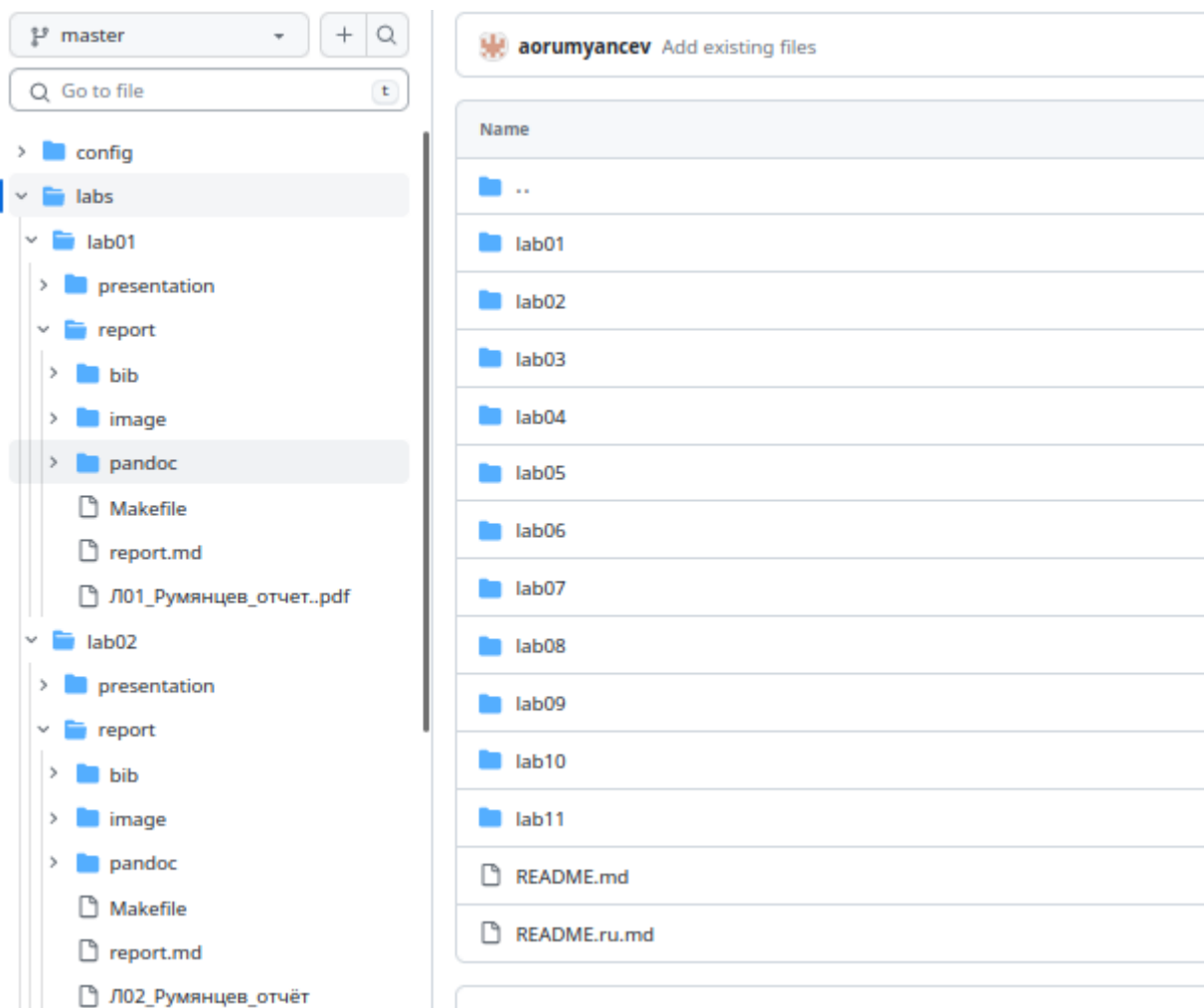


Рисунок 34

Вижу, что отчеты по лабораторным работам находятся в соответствующих каталогах репозитория (рис 35,36)

aorumyancev Add existing files		
fc8f244 - 21 minutes ago History		
Name	Last commit message	Last commit da...
..		
bib	feat(main): make course structure	2 days ago
image	feat(main): make course structure	2 days ago
pandoc	feat(main): make course structure	2 days ago
Makefile	feat(main): make course structure	2 days ago
report.md	feat(main): make course structure	2 days ago
Л01_Румянцев_отчет..pdf	Add existing files	21 minutes ago

Рисунок 35

aorumyancev Add existing files		
1126aac - 13 minutes ago History		
Name	Last commit message	Last commit da...
..		
bib	feat(main): make course structure	2 days ago
image	feat(main): make course structure	2 days ago
pandoc	feat(main): make course structure	2 days ago
Makefile	feat(main): make course structure	2 days ago
report.md	feat(main): make course structure	2 days ago
Л02_Румянцев_отчёт	Add existing files	13 minutes ago

Рисунок 36

5.Выводы

При выполнении данной лабораторной работы я изучил идеологию и применение средств контроля версий, а также приобрел практические навыки по работе с системой git

6.Список литературы

- 1. Архитектура ЭВМ**
- 2. Git – gitattributes Документация**